

Trabajo Fin de Grado

Control de formaciones multirobot con visión

Multirobot formation control with vision

Autora

Begoña Pascual Navarro

Director

Eduardo Montijano Muñoz

Ponente

Carlos Sagüés Blázquez

Escuela de Ingeniería y Arquitectura
2016



DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D^a. Begoña Pascual Navarro
con nº de DNI 25209449 T en aplicación de lo dispuesto en
el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo

de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la

Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)

Grado en Ingeniería Electrónica y Automática

(Título del Trabajo) Control de formaciones multirobot con visión

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada
debidamente.

Zaragoza, 24 de Junio de 2016

Fdo: _____

Agradecimientos

En mi primer lugar, agradecer a Eduardo Montijano todo el apoyo y dedicación que me ha dado durante estos últimos meses. Tu ayuda y confianza han sido imprescindibles para la realización de este trabajo. También a Carlos Sagüés, por todos tus consejos no solo durante este último año. Gracias a tu “mochilica” rosa, todo ha sido más ameno.

A mi hermana, juntas hemos compartido muchos años de estudio y no iba a ser menos durante los meses de realización del trabajo de fin de grado. Gracias por apoyarme en todos los aspecto de mi vida.

A mis padres, vosotros me habéis enseñado que con constancia y un poco de ganas se puede conseguir cualquier cosa.

A Fer, por aguantar mis cambios de humor durante estos últimos meses. Gracias por hacerme ver que todo no es tan negro como lo puedo llegar a pintar. Siempre sabes sacarme una sonrisa.

A Sofía, juntas nos metimos en este lío y juntas vamos a decir que hemos salido. Solo puedo decir que este verano vamos pasarlo como nunca, no me defraudes cari.

A mis amigas, que no compañeras, de la universidad. Elena, Irene y Marta, estos años han sido increíbles gracias a vosotras. Habéis sido amigas, compañeras de trabajos, de prácticas y confidentes de fiestas. Juntas hemos formado un magnifico equipo, espero que sigamos así por mucho tiempo.

¡Gracias a todos!

RESUMEN

CONTROL DE FORMACIONES MULTIROBOT CON VISIÓN

La idea de robots autónomos que se desplazan en un entorno desconocido de forma automática plantea un sinnúmero de ventajas para los seres humanos. Contar con un sistema multirobot permite realizar tareas de forma segura y eficiente. En este contexto, se deben asegurar las estrategias de coordinación entre robots, con el fin de conseguir que optimicen la percepción del entorno. Sin embargo, hay que tener en cuenta que el aumento del número de robots que participan en la tarea, complica de manera significativa la ejecución de la misma debido a las restricciones de movimiento de los robots.

El objetivo principal de este proyecto es llevar a cabo el movimiento en formación de un sistema multirobot en un entorno real. Para ello, se cuenta con un algoritmo de control de formaciones multirobot desarrollado en un proyecto fin de carrera [1] que resuelve la tediosa tarea del aumento de variables a controlar con el aumento del número de robots.

Este algoritmo proporciona para cada uno de los robots que componen el sistema unas velocidades en unidades de m/s y rad/s, en caso de velocidad lineal y velocidad angular respectivamente. Dichas unidades no son representativas para los robots, por lo que se ha realizado una conversión de velocidades del algoritmo a velocidades de los motores de los robots.

Llevar a la realidad un código, que antes estaba implementado en simulación, trae consigo numerosos problemas que obviamente antes no han surgido pero que ahora habrá que tratar. En segunda instancia, para solventar dichos problemas, se ha desarrollado un sistema de reconocimiento por visión de los robots para poder estimar la posición real de cada uno y conseguir la formación deseada. Para ello, se propone el tratamiento de un algoritmo que permite la identificación de los distintos robots usando una cámara área. El algoritmo elegido ha sido el reconocimiento por colores, colocando unas etiquetas identificadoras a cada robot. Este algoritmo considera las restricciones de visión que se plantean, como la necesidad de conocer la orientación y posición de cada uno de ellos y la percepción de todos los robots dentro del mapa de trabajo.

Finalmente, se han realizado diferentes experimentos para observar el comportamiento de los robots y se ha comprobado que el principal objetivo del trabajo, conseguir el movimiento en formación de robots reales, ha sido logrado.

Índice general

Capítulo 1. Introducción.....	9
1.1 Introducción.....	9
1.2 Objetivos	10
1.3 Alcance.....	10
1.4 Contenido de la memoria.....	11
Capítulo 2. Control de robots reales en formación.....	13
2.1 Movimiento de robots reales en formación	13
2.2 Descripción de los robots reales	14
2.3 Conversión de la velocidad.....	15
2.4 Resultados experimentales.....	17
Capítulo 3. Localización por visión	20
3.1 Descripción del problema	20
3.2 Identificación de los robots: detección por color	21
3.3 Resultados experimentales.....	24
Capítulo 4. Conclusiones finales	27
4.1 Conclusiones	27
4.2 Líneas futuras.....	27
Bibliografía	28
Anexos	29
Anexo A. Movimiento de robots reales en formación	29
Control por Particiones de Voronoi	29
Anexo B. Conversión de velocidad.....	32
Anexo B.1 Ensayos de velocidad de simulación a comandos del robot.....	32
Anexo B.2 Análisis de velocidades de comando a motor derecho e izquierdo.....	37
Anexo C. Resultados Experimentales.....	38
Experimento 2 con identificación visual	38

Capítulo 1. Introducción

1.1 Introducción

Desde siempre, la humanidad ha intentado mejorar la vida de las personas realizando tareas cotidianas de manera automática. Trabajar con sistemas multirobot resulta muy ventajoso, ya que un único robot permite realizar cualquier tarea de manera más eficiente, rápida, segura y robusta, colaborando en la manipulación, transporte de objetos o seguridad de recintos.

Para el desarrollo del proyecto se ha partido de dos artículos de investigación para el control de sistemas multirobot, [2] y [3], y de un proyecto fin de carrera [1]. Los dos artículos de investigación han servido de referente para comprender la relación entre la formación de robots y su control por visión. En uno de los artículos se aborda el problema del control visual de un conjunto de robots móviles [2]. En el otro se describe un nuevo método de control basado en visión para impulsar un conjunto de robots que se mueven a una formación deseada [3]. En cuanto al proyecto fin de carrera, se ha utilizado el control desarrollado en ese proyecto, basado en particiones de Voronoi [1], para llevar a la realidad lo desarrollado en simulación.

Este trabajo pretende adaptar el control por particiones de Voronoi a un entorno real con robots reales. Este control logra que todos los miembros del sistema optimicen la percepción del entorno reduciendo el problema de control únicamente a tres variables, centro (x, y) y radio. Para conseguir el movimiento de los robots reales se ha realizado una conversión de velocidades de simulación a velocidades entendibles para los motores de cada uno de los robots.

Sin embargo, dado que nos encontramos en un entorno real existen limitaciones que no aparecen en simulación tales como restricciones de movimiento de los robots, errores en la posición, rugosidades del suelo, saturaciones, velocidades útiles de los robots reducidas a un rango, etc. Estas limitaciones provocan que el movimiento de los robots reales no se corresponda con el generado por la simulación pese a las conversiones de velocidad realizadas.

Dada esta situación, es necesario incluir algún tipo de sensor que estime la posición de cada robot y obtener de esta manera unas velocidades acordes a las posiciones reales en las que se encuentran los dispositivos. El modo elegido para llevar a cabo esta percepción del sistema multirobot ha sido la visión por computador.

Para el reconocimiento de cada uno de los robots, se ha utilizado el reconocimiento de patrones. Esta técnica consiste en la extracción de unas características de la imagen y comprobar si dichas características se encuentran en una imagen cuya información es conocida. Para la identificación de los robots, se ha colocado una cámara área y se ha definido un mapa de trabajo. Se barajaron algunas opciones para el reconocimiento de los miembros del sistema multirobot y finalmente la opción elegida fue a partir de unas etiquetas de colores colocadas en la parte

superior de cada uno de los robots. Esta técnica ha resultado muy eficiente dadas las características del ambiente.

Por tanto, este trabajo pretende unir las ventajas que ofrecen el movimiento multirobot con el reconocimiento de patrones a través de una cámara aérea.

1.2 Objetivos

El objetivo principal de este proyecto es llevar acabo el movimiento en formación de un sistema multirobot en un entorno real con robots reales. Para ello se cuenta con un proyecto fin de carrera [1] en el que se simulaba formaciones multirobot a partir de control por particiones de Voronoi. Por lo tanto, el objetivo es conseguir el mismo movimiento en formación de robots reales en una plataforma real. Para ello, se plantean dos objetivos más concretos:

- Dado que en el control en simulación genera unas velocidades en m/s y rad/s para conseguir el movimiento de la formación y los robots reales responde a valores adimensionales comprendidos entre -255 y 255, uno de los principales objetivos es relacionar las unidades de velocidad entendibles para el ser humano con el rango de velocidades adimensionales en las que se mueven los robots. De este modo, se conseguirá un movimiento aproximado entre las velocidades generadas por el control y el movimiento que describa el robot.
- En simulación, todo tipo de movimiento es posible y alcanzable, sin embargo, para un robot real está situación ideal no es posible por lo que el movimiento de simulación no se corresponde con el real, por tanto, será necesario realizar una identificación de cada uno de los robots. El segundo objetivo es desarrollar un algoritmo que permita la percepción de los robots por visión a partir de una cámara aérea.

1.3 Alcance

Se parte de un proyecto fin de carrera [1] que proporciona simulaciones de movimientos de robots en formación en el entorno de MATLAB. Por lo tanto, esta ha sido la herramienta utilizada para desarrollar este trabajo. Además, en MATLAB está integrada la toolbox Computer Vision System Toolbox™ que proporciona algoritmos, funciones y aplicaciones para el diseño y simulación de sistemas de visión por ordenador y procesamiento de vídeo.

En la Figura 1 se observa el esquema de control del que se ha partido. El bloque corresponde al control implementado en el proyecto fin de carrera [1], control por particiones de Voronoi.

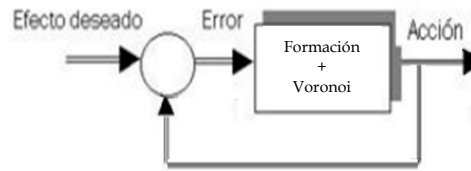


Figura 1. Bucle abierto de control.

Sin embargo, para un robot real, la situación de idealidad de movimientos que se ofrece en simulación no es posible lo que deriva en un movimiento en formación erróneo.

Se pretende conseguir el desarrollo de un sistema capaz de mover robots reales con un movimiento similar al que se propone en las simulaciones. Para ello, se plantea la necesidad de ciertas transformaciones de velocidades de simulación a velocidades que permitan mover los motores de cada uno de los robots. Además de un sistema de localización que permita identificar la posición de los robots en cada instante y lograr así una correcta formación.

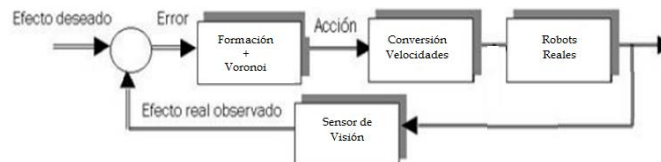


Figura 2. Esquema de control en bucle cerrado básico.

En el esquema de control de la Figura 2 se identifica visualmente cada uno de los elementos que compondrá el bucle cerrado de control:

- Formación + Voronoi: control por particiones de Voronoi. A alto nivel precisa de centro y radio y calcula velocidad lineal y angular necesaria para cada uno de los robots.
- Conversión de velocidades: las velocidades generadas por el control son en m/s y rad/s, por lo que antes de ser transmitidas a los robots tienen que ser convertidas a unidades significativas para cada uno de ellos.
- Robots reales: sistema multirobot de N robots reales. Realizan el movimiento con las velocidades transformadas.
- Sensor de visión: cámara aérea encargada de capturar una imagen de la escena en la que los robots se encuentran para extraer la posición y orientación en la que se encuentran cada uno ellos. Su papel es imprescindible puesto que permite llevar el control a un entorno real cerrando el bucle de control.

Finalmente se adaptará una plataforma real para el control y se propondrán diversos trayectos para comprobar el funcionamiento.

1.4 Contenido de la memoria

El resto de la memoria se estructura de la siguiente manera:

En el Capítulo 2 se expone el algoritmo de navegación de los robots reales, la descripción de los robots reales que se van a utilizar y se plantean las conversiones de velocidad. Además, se realiza un experimento para comprobar el funcionamiento de dichas conversiones.

En el Capítulo 3 se presenta la localización de los robots en posición y orientación. También se realizan varios experimentos para analizar el nuevo comportamiento de la formación teniendo información de la posición real de los robots.

En el Capítulo 4 se resumen las conclusiones extraídas de este trabajo y se describen las líneas de trabajo futuro.

Finalmente se completa la información en la bibliografía y anexos.

Capítulo 2. Control de robots reales en formación

En este capítulo se explica el tipo de control propuesto, cómo son los robots reales con los que se va a trabajar, la conversión de velocidades de simulación a velocidades de los motores del robot y los resultados experimentales obtenidos con dichos robots en una plataforma real. El método completo se describe el Anexo A.1

2.1 Movimiento de robots reales en formación

En primer lugar se describe brevemente el algoritmo utilizado para la navegación del equipo de robots. Esta técnica se propuso en [1].

A medida que aumenta el número de robots de nuestro sistema, se incrementa el número de variables que definen el espacio de estados del conjunto. Cada robot adicional introduce una nueva posición y orientación a controlar. Este aumento de variables presenta un problema a la hora de conseguir un algoritmo de control eficaz para todo el conjunto de dispositivos.

De esta problemática surge la idea de crear un nuevo espacio, una abstracción cuya dimensión sea constante e independiente del número de robots que formen el sistema. Esta planificación plantea reducir el número de variables a controlar en el sistema de navegación simplemente a tres: las coordenadas x e y de un círculo y el radio del mismo. Estas tres variables son por tanto independientes del número de robots de nuestro sistema, los cuales se encuentran distribuidos en esa área.

Para conseguir el movimiento de los robots dentro de la abstracción, se subdivide el círculo en particiones, tantas como robots existan. El algoritmo elegido para controlar nuestro sistema multirobot es el control por particiones de Voronoi. Podría haberse elegido otro tipo de control como el control mediante abstracción geométrica, sin embargo, fue descartado puesto que presentaba menos ventajas frente al control mediante particiones de Voronoi. Internamente, en el algoritmo de particiones de Voronoi, cada robot es movido dentro de su región hacia el centroide de la misma. De este modo se consigue el movimiento en formación. Este proceso se resume en la Tabla 1:

<ul style="list-style-type: none">> <i>requiere centro y radio;</i>> <i>mientras error repetir</i><ul style="list-style-type: none">> <i>Calcular posición de la abstracción</i>> <i>Calcular las nuevas particiones de Voronoi</i>> <i>Para cada región de Voronoi, calcular un objetivo (centroide)</i>> <i>Calcular velocidades</i>> <i>Movimiento de los robots</i>> <i>Actualizar centro y radio</i>

Tabla 1. Pseudocódigo del bucle abierto de control mediante particiones de Voronoi.

En la Figura 3, se muestra un ejemplo de partición de Voronoi para un entorno con forma de circunferencia y 3 robots. Se puede ver como en color cian se muestran los centroides de la región asociada a cada robot y en azul la circunferencia que representa a la abstracción y las líneas que parten las áreas de movimiento de cada robot.

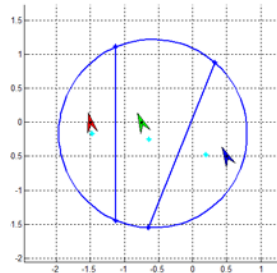


Figura 3. Partición de Voronoi con 3 robots.

Este tipo de control evita colisiones entre los robots, lo cual es una ventaja a la hora de llevarlo a un entorno real y además genera un reparto de los robots homogéneo dentro de la abstracción. Por último, el uso de Voronoi implica que cada robot determina su zona de movimiento conociendo únicamente las posiciones de sus vecinos y la de la abstracción, sin ser necesario datos como la situación de todos los dispositivos.

Una vez definido el tipo de control que va a regir en nuestro sistema multirobot y sus ventajas, vamos a proceder a la descripción física y estructural de los robots utilizados.

2.2 Descripción de los robots reales

Los robots empleados en este trabajo se corresponden con los que se muestran en la Figura 4:

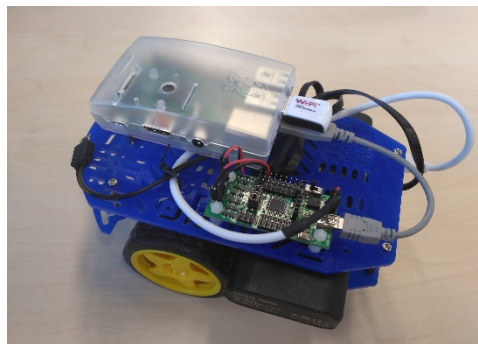


Figura 4. Robot de trabajo.

Nos encontramos con N robots terrestres de tracción diferencial más un tercer apoyo que les dota de orientación. Constan de una tarjeta Dagu Arduino Mini Driver para el control de servos y motores del robot (para la rueda derecha y la rueda izquierda), una placa Raspberry Pi 2 y un USB powerbank para la alimentación [6].

Son robots teleoperados, ya que se trata de robots que se controlan a distancia mediante cable inalámbrico, en nuestro caso Wi-Fi desde un ordenador. El hecho de tener una conexión Wi-Fi en nuestro equipo nos aporta ventajas como: ausencia de cables que podrían dificultar el movimiento de los robots y conexión desde cualquier

lugar. Una vez creada la señal, podemos establecer comunicación con nuestro sistema multirobot.

Desde el punto de vista de la comunicación, cada robot es un terminal IP que permite comunicación través de una red IP gracias a la red Wi-fi creada anteriormente. El patrón que sigue la IP de nuestros dispositivos es $192.168.173.10X$ donde X es el número que identifica a cada robot. Podemos encontrar este número en la parte lateral derecha la placa Raspberry.

Los comandos de velocidad se envían con la función:

```
glnSendrobot (t, vR, vL, comando);
```

con t correspondiente a la dirección IP, vR y vL velocidad rueda derecha y velocidad rueda izquierda respectivamente y comando el modo de trabajo del robot. El rango de velocidades de las ruedas está limitado entre -255 y 255. A bajo nivel, el control por particiones de Voronoi genera una velocidad en m/s para velocidad lineal y rad/s para velocidad angular. Por tanto, estas unidades no son comprensibles para nuestro sistema multirobot real con lo que el primer problema radica en convertir las velocidades de m/s y rad/s a unos comandos velocidad a los que los motores derecho e izquierdo de cada uno de los robots puedan responder correctamente.

2.3 Conversión de la velocidad

Como se ha comentado en el apartado anterior, los robots responden a comandos de velocidad limitados entre 255 y -255. En este apartado vamos a explicar todos los experimentos realizados para la conversión de velocidades.

Para relacionar las velocidades generadas en la simulación con las adimensionales, se han mandado distintos comandos de velocidad a distintos tiempos y ha medido el desplazamiento realizado.

En el caso de velocidad lineal, se ha obtenido una relación de velocidades en m/s a velocidad de comandos, 2.1 y 2.2. Ambas son ecuaciones similares puesto que son lineales. Estas expresiones son válidas para el rango de velocidad comprendido entre 0,01 y 0,05 m/s. Si las velocidades son superiores o inferiores, las expresiones no son válidas. Para velocidades inferiores a 0,01 m/s el valor de comando asignado es 0. Para superiores a 0,05 m/s el valor de velocidad de comando asignado es el máximo, 255.

En el caso de velocidad angular se han obtenido de nuevo dos expresiones, 2.3 y 2.4, una para velocidades angulares positivas y otra para velocidades angulares negativas. Dichas expresiones son válidas cuando las velocidades se encuentran entre 0,8 y 4,8 rad/s. Si las velocidades son superiores o inferiores a estos valores, las expresiones no son válidas. Para velocidades inferiores a 0,8 rad/s el valor de comando asignado es 0. Para superiores a 4,8 rad/s el valor de velocidad de comando asignado es 255.

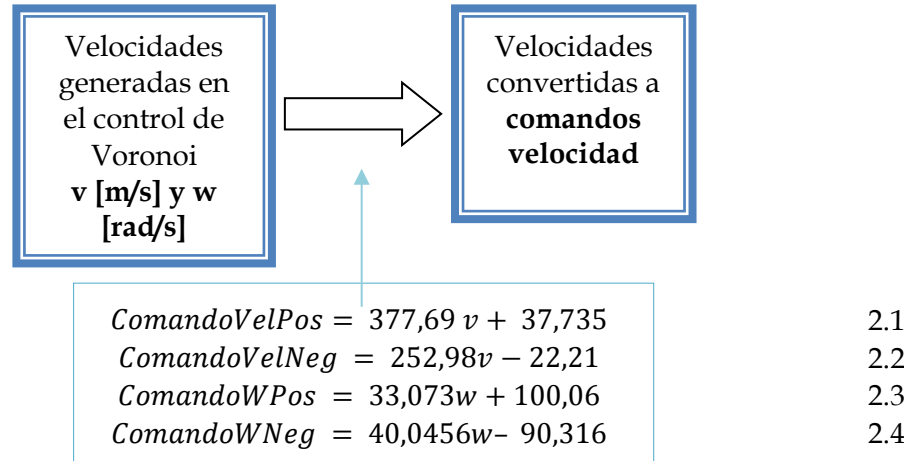


Figura 5. Esquema de conversión de velocidades de control al robot.

Para el desarrollo completo de los ensayos acudir al Anexo B.1.

En cuanto a la velocidad transmitida a las ruedas, se observan problemas de no linealidades de movimiento derivado de los rangos en los que la velocidad es nula, muy baja o muy alta. Dada esta situación, ha sido necesario realizar un ajuste proporcional de velocidad lineal y velocidad angular para que los robots describieran correctamente sus movimientos. Por ello, se han implementado unas constantes proporcionales a v y w dependientes de la velocidad en cada instante:

$$(v, w) = f(k_v, k_w) \quad 2.5$$

De este modo, dados unos valores límites estudiados experimentalmente de velocidad lineal y angular (e_v y e_w) utilizados para discriminar entre movimiento rotacional, movimiento traslacional, movimiento a baja velocidad y movimiento a alta velocidad. Estos valores son independientes para cada uno de los robots y se han considerado diferentes situaciones. Dicho análisis puede consultarse en el Anexo B.2. En la Figura 6, se observan las expresiones que dan valor a las velocidades del motor derecho y del motor izquierdo además de las constantes de proporcionalidad que irán adoptando diferentes valores dependiendo del tipo de movimiento:

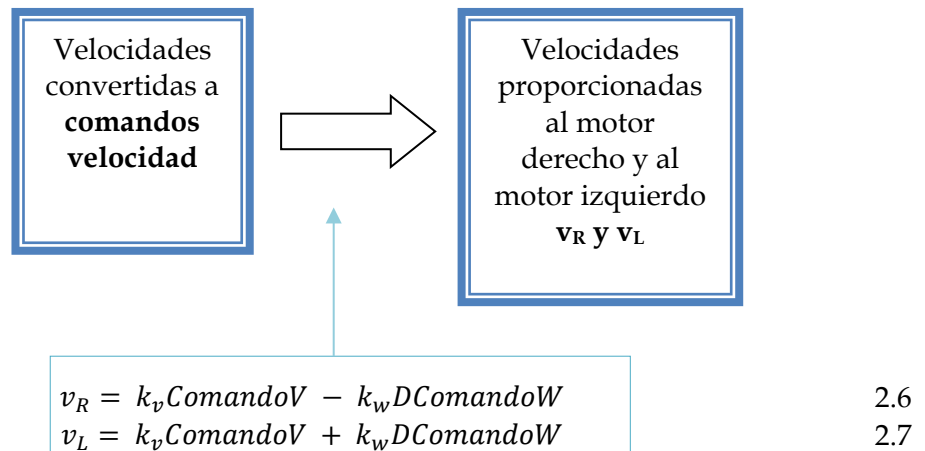


Figura 6. Esquema de conversión entre velocidades adimensionales y velocidades de motor derecho e izquierdo con $D = 0,05\text{cm}$

2.4 Resultados experimentales

A continuación, se muestra un ejemplo de control considerando una abstracción en forma de círculo. Se representan las configuraciones iniciales y finales con sus correspondientes particiones de Voronoi, las acciones aplicadas a los robots y los errores.

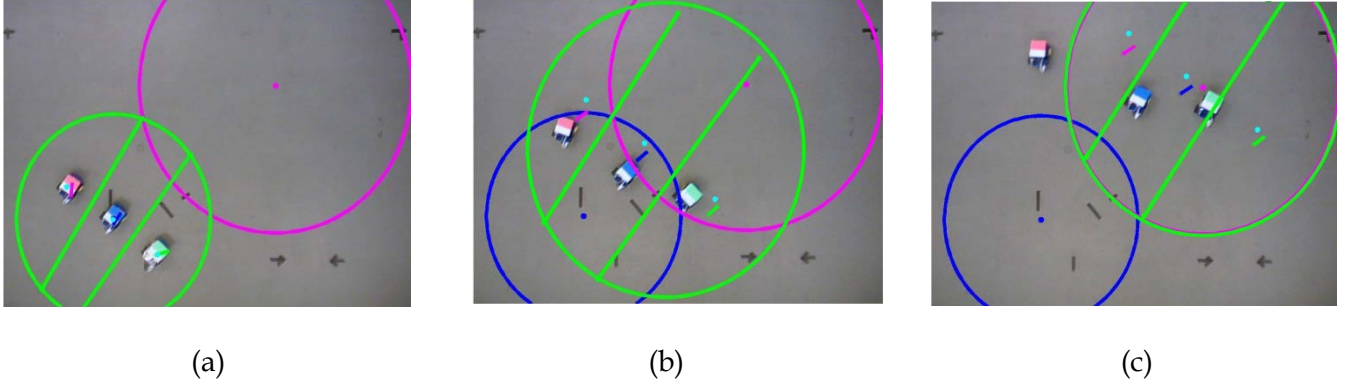


Figura 7. (a) Posición inicial y objetivo para 3 robots. (b) Desplazamiento de los robots. (c) Posición final.

En Figura 7 (a) se muestra una abstracción formada por 3 robots situados aleatoriamente. Se ha seleccionado una configuración final de centro (2m, 1,5m) y radio 1m. En azul, debajo de la verde, encontraríamos la abstracción inicial, en verde el cálculo inicial de las particiones antes de comenzar a moverse y en magenta la abstracción final. En cian los centroides a los que deben dirigirse cada robot. Las líneas magenta, azul y verde representan el vector de orientación y posición en los que deberían encontrarse cada uno de los robots según el control en simulación.

Las velocidades generadas por el control son convertidas y enviadas a cada uno de los robots. Observamos cierta coherencia en el movimiento, sin embargo, el desplazamiento real de los robots con el simulado no es el mismo, Figura 7 (b). Se aprecia como la simulación genera unas posiciones a partir de unas velocidades que sí que pueden cumplir los robots en simulación, pero no los robots reales. Se observa una pequeña separación entre la posición de simulación y la de los robots reales.

En la Figura 7 (c) se observa como todos los robots reales se acercan a la posición deseada, sin embargo, ha incrementado su error con respecto a la posición generada por el control. Uno de ellos incluso ha abandonado la abstracción y otro ha invadido la partición de su vecino.

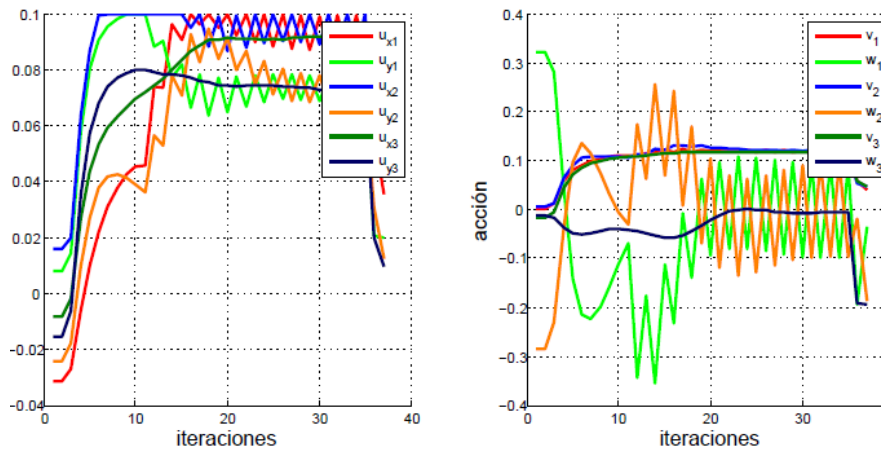


Figura 8. Acción para $N = 3$ centro = $(2m, 1,5m)$ y radio = $1m$.

En cuanto a la acción aplicada a cada uno de los dispositivos, se muestran las acciones respecto a los ejes absolutos y las acciones lineal y angular en la Figura 8. En la gráfica de izquierda se puede ver como se produce un aumento de la acción progresivo hasta que alcanza el valor máximo establecido en 0,1. La acción es casi máxima debido a que la separación entre el centroide a dirigirse y la posición del robot supera un cierto valor. En la gráfica de la derecha, se observa algo similar. La acción angular de los robots magenta y verde, empieza a oscilar entorno a 0 a partir de la iteración 20. Esto se debe a que se están produciendo pequeños cambios en la orientación. También se observa como la acción de lineal es casi constante dado que el desplazamiento ha sido prácticamente lineal.

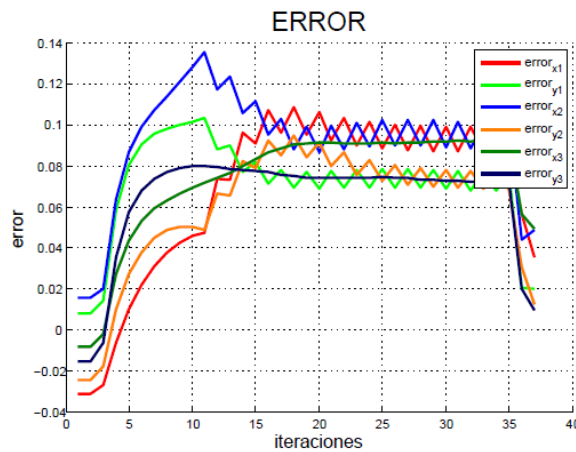


Figura 9. Error de posición en x e y para cada robot de simulación con respecto al centroide.

En la Figura 9 se aprecia como los errores de posición no superan los 0,14 metros entre la posición de la simulación de cada robot con su centroide. Dado que el desplazamiento ha sido de unos dos metros el error máximo está entorno al 8% por lo que se puede considerar relativamente bajo.

Si se compara la separación que existe entre el centroide y la posición de cada uno de los robots reales, se observa como la distancia es mucho mayor, llegando en el caso del robot rosa a unos 80cm. Esto supone un error máximo entorno al 40%.

Analizando los resultados obtenidos, se puede afirmar que el movimiento real no se corresponde con el simulado: existen errores de posición, robots reales fuera de la abstracción, posibles colisiones al invadir particiones vecina, etc. En la simulación, está planeado que el movimiento de la abstracción sea acorde al de los robots en casos en los que todo movimiento es alcanzable. Sin embargo, para un robot real esta situación ideal no es posible con lo que cabe la posibilidad de que los robots salgan de la abstracción.

Las causas a las que puede deberse dichos errores son: rugosidades del suelo, problemas asociados a saturaciones comentados en la otra sección, ajuste empírico no totalmente perfecto, problemas de localización, etc. Los robots se encuentran en un entorno real, con lo que era de esperar que su movimiento no se correspondiese con el simulado.

El siguiente capítulo trata de poner solución a todos estos errores utilizando un sistema de visión para estimar la posición real de cada uno de los miembros de nuestro sistema multirobot.

Capítulo 3. Localización por visión

3.1 Descripción del problema

La identificación de cada robot es un aspecto básico para lograr la correcta disposición del sistema multirobot. Para ello, será necesario identificar posición y orientación de cada uno de ellos, es decir, necesitamos de un algoritmo que nos permita reconocer de manera global a todos los robots.

Existen diferentes métodos para conocer la posición. Podemos clasificarlos en dos tipos:

- Estimadores explícitos: como por ejemplo la odometría...
- Estimadores basados en la percepción del entorno: que pueden ser basados en técnicas de emparejamiento o haciendo uso de sensores, como puede ser un sónar, un láser, una cámara...

La alternativa elegida para la estimación de la posición de cada uno de los robots es dotar a los dispositivos de un sistema sensorial capaz de proporcionar suficiente información del entorno como para que puedan de forma autónoma inferir su localización. En nuestro caso, la solución adoptada es una cámara que tiene percepción de todo el mapa por el que se van a mover los robots. Esta cámara área se encuentra a bastante altura para poder abarcar la situación de todos los robots. Se dispone de una zona de trabajo habilitada para la cámara cuyo campo visual es de 3x2 metros en largo y ancho.

Las cámaras de vídeo son sin duda uno de los sensores más utilizados en percepción gracias a las numerosas ventajas que presentan: alta resolución, velocidad en la adquisición de la información, bajo consumo, no necesitan emplear dispositivos mecánicos y son relativamente baratas.

Se plantean diferentes opciones para su reconocimiento visual. La opción elegida debe estar dotada de diferentes características tales como localizar independientemente a cada robot y tener capacidad de procesamiento de las imágenes a cierta distancia de los robots.

En resumen, el objetivo pretendido en este capítulo es desarrollar un método general y práctico para la identificación de los robots a partir de imágenes captadas por una cámara aérea instalada en el techo.

3.2 Identificación de los robots: detección por color

En esta sección se plantea como se va estimar la posición y orientación real en la que se encuentra cada uno de los miembros de nuestro sistema multirobot.

La posición y la orientación es sin duda uno de los aspectos más importantes en la navegación automática de un vehículo ya que nos permite monitorizar la ejecución de los movimientos para llegar al destino deseado. Además, atendiendo a los problemas obtenidos en el movimiento en bucle abierto, se ha limitado el movimiento de la abstracción, es decir, la actualización de radio y centro de acuerdo a las restricciones del movimiento real de los robots. Por lo tanto, cuando cualquiera de los robots se encuentre muy cerca de la abstracción se paraliza la actualización de la misma hasta que el robot o los robots estén reubicados.

Se necesita conocer información de cada robot, por tanto, es necesario identificar cada robot y diferenciarlo de los demás para poder conocer su posición y orientación. En una etapa inicial se estudiaron varias posibilidades para realizar el reconocimiento por visión de los robots. Se hicieron algunos ensayos utilizando descriptores invariantes, como SURF [7], sin embargo, dadas las particularidades de localizar muchos robots, es necesario que la cámara esté situada a cierta altura, con lo que al intentar reconocer un reducido número de píxeles que ocupa cada robot en la imagen la opción fue descartada por ser poco robusta.

La otra posibilidad estudiada en la identificación por color de cada uno de los robots. Para el desarrollo de este método es necesario recordar que necesitamos conocer orientación y posición, por lo que si las etiquetas son para todos los robots de la misma forma, no podemos dotarlos de un único color, sino que son necesarios dos colores que permitirán conocer su orientación como se puede ver en la Figura 10. A partir de los centroides de cada etiqueta se calcula la orientación en la que se encuentra cada uno de los robots:

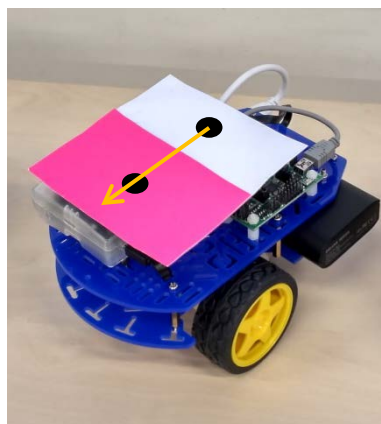


Figura 10. Orientación de robot real. El vector que une ambos centroides proporciona la orientación del robot.

Dotando de dos colores a cada robot podemos obtener un vector de orientación y posición. El número de colores necesarios será por tanto:

$$\text{Número de Colores} = 2N \quad 3.1$$

considerando N el número de robots implicados en la tarea de planificación. Para hacerlo más eficiente computacionalmente, se tomó la decisión de dotar a los robots de un color identificativo a cada uno (magenta fosforito, verde fosforito y azul fosforito) y otro común a todos (blanco):

$$\text{Número de colores} = N + 1 \quad 3.2$$

De esta manera, el proceso de segmentación de cada uno de los colores se reduce en gran medida. En la Figura 11 se muestra a los tres robots con sus respectivas cartulinas:

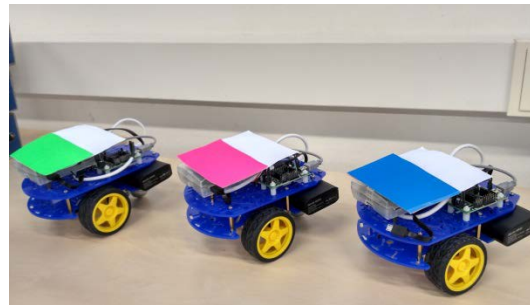


Figura 11. Robots con cartulinas identificadoras.

El modelo de color elegido es el modelo RGB (Red, Green, Blue) [4], en él se representa el color mediante la mezcla por adición de los 3 colores primarios. La intensidad de cada componente se mide según una escala que va del 0 al 255.

Para la identificación de cada uno de los colores se ha realizado un proceso de muestreo de píxeles pertenecientes a cada uno de los colores de cada robot. Se ha calculado los valores medios y desviaciones de las tres componentes espectrales RGB en cada tono, es decir, los valores medios y desviaciones de rojo, verde y azul en cada una de las etiquetas. Manteniendo las condiciones de luminosidad y sombra del entorno similares a las de la calibración de los colores, no será necesario volver a evaluar los colores elegidos. Esto evita la tediosa tarea de introducir a mano el posible valor de los colores y se asegura que los rangos de valor medio y desviaciones son correctos y van a permitir la identificación de los colores con total eficiencia y robustez.

Una vez etiquetados los robots, se describe el proceso de detección de los mismos. En primer lugar se obtiene una captura de la escena con la cámara aérea, Figura 12. Posteriormente, la imagen es segmentada buscando reducirla a los datos de interés con la información que nos importa. En nuestro caso, lograr una imagen binaria que muestra los colores elegidos en color blanco y el fondo negro, preservando la información estructural importante en la imagen, Figura 12:

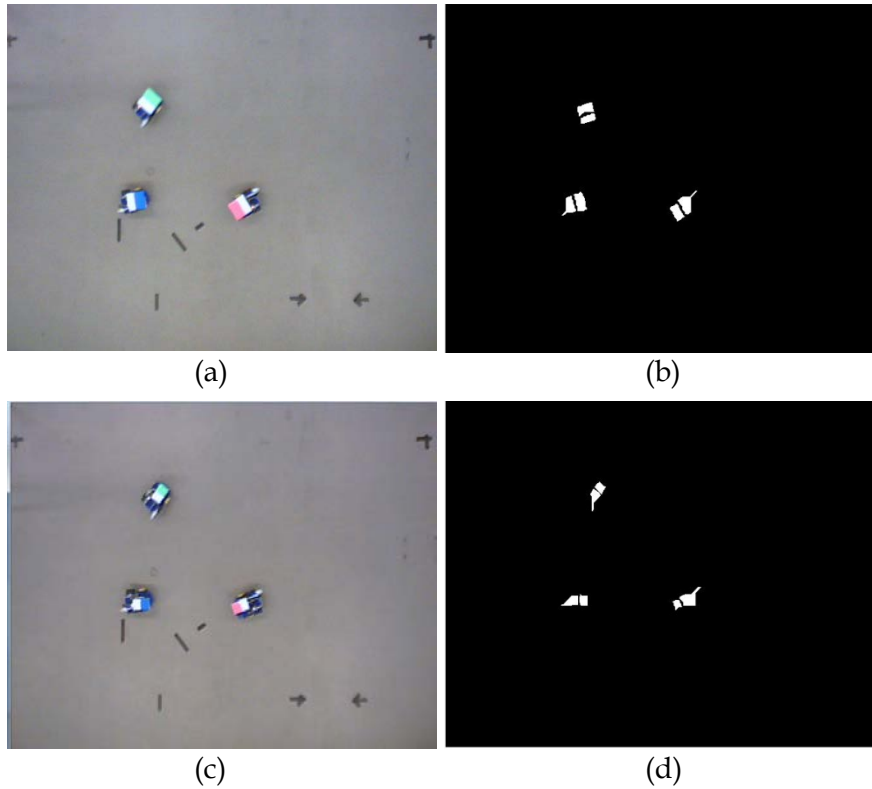


Figura 12. Captura de la escena (a) y (c) y segmentación de la misma (b) y (d) respectivamente. Las imágenes de la zona inferior son etiquetas más pequeñas que las superiores, se observa que los robots siguen siendo localizados.

El siguiente paso consiste en extraer las características de la imagen segmentada. De este proceso se obtiene un conjunto de coordenadas de pixel que representan cada uno de los colores encontrados en la imagen (el centroide). Se obtendrán un total de $2*N$ centroides. Y finalmente, se procede a la identificación de cada uno de los robots. Se analiza el área de la segmentación de los colores encontrados debido a que por problemas de sombras o diversos ruidos puede que se identifiquen zonas que no corresponden a las etiquetas de los robots. Estos errores se suponen de área pequeña y, por tanto, serán descartados. Una vez realizada esta selección, se almacena el punto medio de cada pareja de etiquetas, así como la orientación obtenida por el vector que une los dos centroides de dicha pareja. Se considera que dos etiquetas están emparejadas cuando la distancia entre sus centroides es pequeña.

El algoritmo de reconocimiento de color no es todo lo óptimo que se desearía en cuanto a tiempo de procesamiento de imágenes se refiere, entorno a unos 2 segundos por imagen, sin embargo, se ha tener en cuenta que se consigue identificación incluso cuando las etiquetas son pequeñas. Nos ofrece cierto margen de distancia entre cámara y robots. Además, el proceso de calibración de colores, nos proporciona cierta independencia a pequeñas variaciones de luminosidad. Por todo ello, este es el método propuesto para la identificación de cada uno de los robots del sistema multirobot.

3.3 Resultados experimentales

Se han realizado dos experimentos para comprobar el correcto funcionamiento. En esta sección se describe con detalle el primero, mientras que el segundo puede consultarse en el Anexo C. En un primer experimento, se situó a los robots de manera aleatoria. Volvemos a encontrar en azul oscuro la abstracción inicial, en magenta la abstracción objetivo y en verde las particiones calculadas. Se realizó de nuevo para tres robots y se definieron dos objetivos: uno de ellos con centro igual a $(2m, 1,5m)$ y radio $1m$, y el segundo $(1m, 1,5m)$ y radio $0,75m$.

En la Figura 13, se muestra en la imagen (a) la posición inicial de cada uno de los robots. Los puntos de color cian corresponden a los centroides calculados por el control, por lo que en la imagen (b) se observa como cada uno de ellos se ha dirigido a ese objetivo. También se puede ver el vector de posición y orientación sobre cada uno de los robots, dirigido desde el centro de la etiqueta de color hasta el centro de la etiqueta blanca.

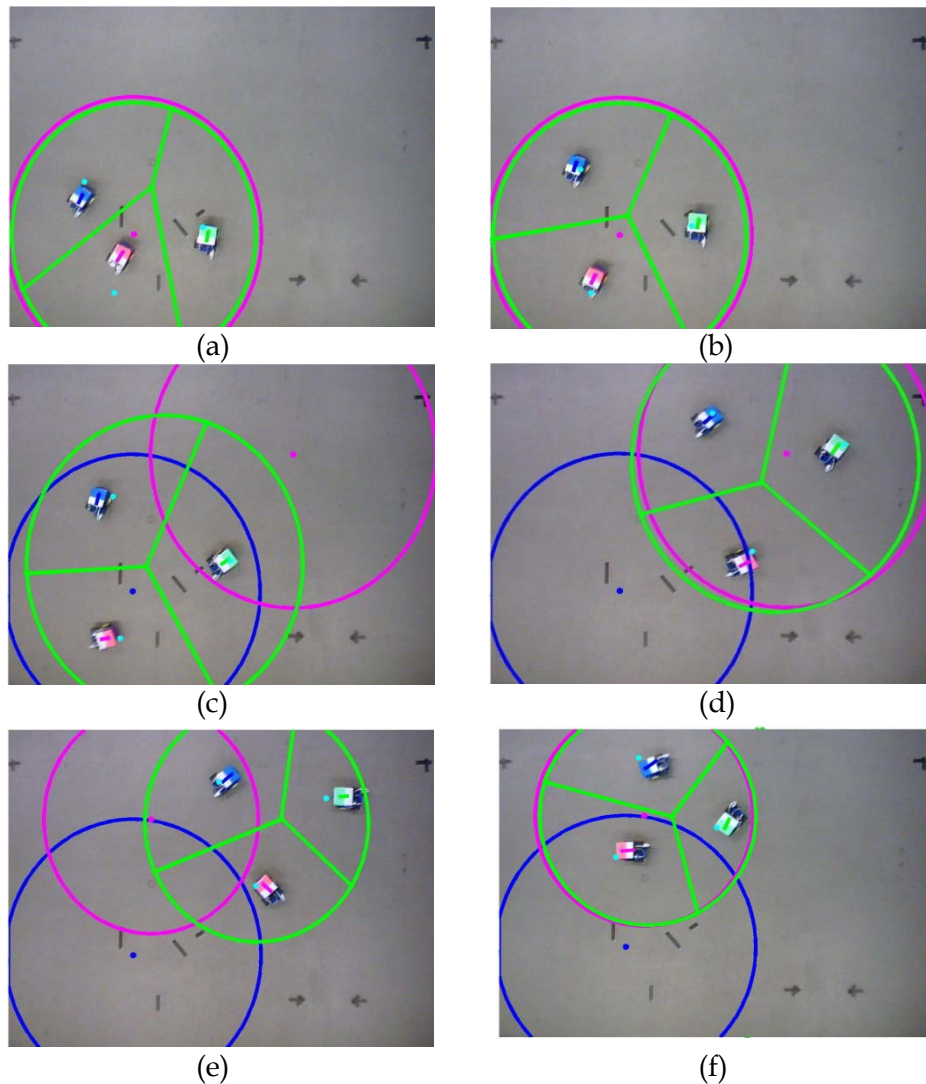


Figura 13. En (a) y (b) reubicación de cada uno de los robots en su objetivo inicial. En (c) y (d) se observa a los robots en bucle cerrado alcanzado primer objetivo. En (e) y (f) los robots en bucle cerrado se dirigen hacia segundo objetivo hasta finalmente alcanzarlo.

A continuación, la partición avanza a su primer objetivo. Se observa en la Figura 13 (c) y (d) como los errores asociados a posición han desaparecido.

En las imágenes (e) y (f) de la Figura 13, se muestra como los robots se dirigen hacia su segundo objetivo. En la imagen de la izquierda, se puede observar como el robot con etiqueta verde está muy cerca del límite de la abstracción, por lo que la actualización de centro y radio se detuvo hasta que se ha reubicado en una zona un poco más alejada.

También hay que destacar, que la localización de los robots no siempre es perfecta. En la Figura 14, se puede observar como la partición de cada uno de los robots se ha deformado con respecto a la anterior. Este problema se debe a que el robot azul no ha sido correctamente reconocido, se observa como no tiene el vector de orientación y posición sobre la cartulina. El problema ha sido resuelto en la siguiente captura de imagen, y como se ve en la Figura 13 (f), los robots han llegado correctamente a su objetivo sin más complicaciones.

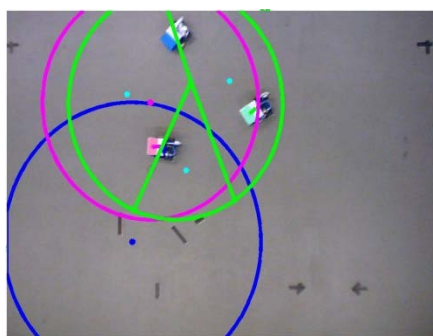


Figura 14. Problemas de localización

Las velocidades, a la vista de las gráficas de la Figura 15, no son muy constantes porque el movimiento es poco preciso. Esto se debe a que dada la realimentación, la corrección de errores es continua, por lo que se producen picos en la acción. Se ve como en la gráfica de la izquierda que los valores de acción no superan en ningún caso los valores de acción máxima, 0,1, como se impuso en la simulación. En la figura de la derecha se observa como las variaciones en la velocidad también son persistentes. Se puede ver como los cambios de velocidad más fuertes se encuentran en las iteraciones cercanas a la llegada del robot al objetivo con el fin de alcanzar una correcta posición y orientación.

En cuanto a la gráfica de la parte inferior, Figura 15, que muestra el error de las posiciones de los robots con respecto a su centroide objetivo, se puede contemplar como rara vez el error es superior a 0,10m. Dado que el movimiento a cada objetivo es de unos 2 metros, nos encontramos con errores del 7%, lo que podemos de calificar de un valor bastante bajo:

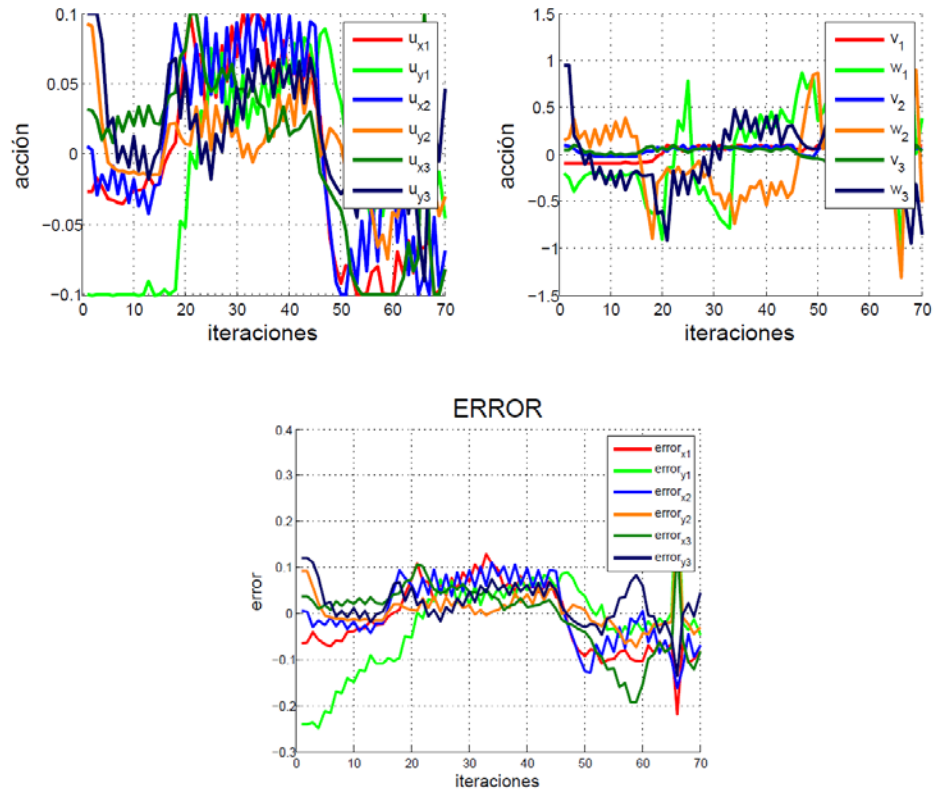


Figura 15. Acción para $N = 3$ centro = $(2m, 1,5m)$ y radio = $1m$ y centro $(1m, 1,5m)$ y radio = $0,75m$. Error de posición en x e y para cada robot con respecto al centroide.

Capítulo 4. Conclusiones finales

4.1 Conclusiones

El proyecto ha tratado de conseguir llevar al mundo real el movimiento en formación de robots reales desarrollado anteriormente en simulación en un proyecto fin de carrera [1]. Para su desarrollo se han tenido en cuenta todas las restricciones que plantea llevar un movimiento simulado a la realidad con las consiguientes perturbaciones del entorno, ajuste empírico no perfecto, limitaciones de movimiento de los robots, etc.

Se ha realizado un ajuste de velocidades para adaptar la simulación a la plataforma real y se ha desarrollado un sistema de identificación visual de cada uno de los robots a partir de etiquetas de diferentes colores.

Con todo esto, se ha logrado conseguir un movimiento en formación bastante aproximado al que debería darse: los robots responden bastante bien a los comandos de velocidad, son correctamente identificados en la mayoría de los casos y además son capaces de describir diferentes formaciones y diferentes movimientos.

Los experimentos han demostrado que es un sistema que responde bien a los cambios realizados y por lo tanto, puede decirse que es un sistema robusto en cuanto a movimiento y reconocimiento de cada uno de los robots se refiere.

4.2 Líneas futuras

Existen puntos en los que no ha sido posible profundizar, que permitirían mejorar los resultados obtenidos en este proyecto. Uno de los principales problemas radica en el tiempo de procesamiento de la imagen obtenida por la cámara aérea. El tiempo de obtención de una nueva posición se encuentra entorno a los 2 segundos, con lo que ralentiza todo el proceso. En el caso de la segmentación por colores, podría reducirse el área de búsqueda de toda la imagen a una zona alrededor del color en cuestión, es decir, realizar tracking de colores. Otra opción podría ser asignar un color a todos los robots, y por emparejamientos, predecir cuál es la posible nueva localización de cada uno de los robots. De esta manera, se podría reducir el número de colores y eliminar calibración muchos colores, con uno sería suficiente.

En el caso del ajuste empírico de las velocidades de los robots reales, podrían realizarse otras transformaciones de velocidades holónomas a no holónomas [8], por ejemplo, realizando otro tipo de ensayos.

Bibliografía

- [1] Vera, D. (2011). Planificación y control con restricciones de formaciones de robots. [Proyecto fin de carrera]. Universidad de Zaragoza. Recuperado el 8 de febrero de 2016, de <https://zaguan.unizar.es/record/6479?ln=es>
- [2] G. Lopez-Nicolas, M. Aranda, Y. Mezouar and C. Sagüés "Visual control for multirobot organized rendezvous," IEEE Trans. Syst., Man, Cybern.B, Cybern., vol. 42, no. 4, pp. 1155–1168, 2012. IEEE Transactions on Systems Man and Cybernetics: Part B.
- [3] M. Aranda, G. López-Nicolás, C. Sagüés, Y. Mezouar "Homography-based multi-robot control with a flying camera," IEEE Transactions on Robotics (ISSN 1552-3098), doi: 10.1109/TRO.2015.2452777, Vol. 31(4): 1064-1071, 2015.
- [4] Richard Szeliski. Computer Vision: Algorithms and Applications. Springer, 2010.
- [5] "Coverage control for mobile sensing networks". Cortés, Jorge; Martínez, Sonia; Karatas, Timur; Bullo, Francesco. 2004. 2, s.l. : IEEE Transactions on Robotics and Automation, 2004, Vol. 20. 243-255.
- [6] Using the Dagu Mini Driver to Build a Raspberry Pi Camera Robot _ Dawn Robotics.https://dl.dropboxusercontent.com/u/63153891/Using%20the%20Dagu%20Mini%20Driver%20to%20Build%20a%20Raspberry%20Pi%20Camera%20Robot%20_%20Dawn%20Robotics%20Blog.pdf
- [7] Bay, H., Tuytelaars, T., & Van Gool, L. (2006, May). Surf: Speeded up robust features. In European conference on computer vision (pp. 404-417). Springer Berlin Heidelberg.
- [8] E. Montijano, E. Cristofalo, M. Schwager, C. Sagües. 2016. Distributed formation control of non-holonomic robots without a global reference frame. IEEE International Conference on Robotics and Automation (ICRA), 5248-5254

Anexos

Anexo A. Movimiento de robots reales en formación

Control por Particiones de Voronoi

Como se ha comentado anteriormente, partimos de un proyecto fin de carrera en el que se obtuvo el control distribuido basado en particiones de Voronoi [5]. Entendemos por control distribuido un control en el que cada robot es capaz de calcular su propia entrada de control sin la necesidad de que exista una unidad de central encargada de todos los cálculos. Este control trata de optimizar la percepción de un entorno cerrado y convexo por un conjunto de robots en un entorno dinámico.

Definimos partición de Voronoi de un entorno Q , como una partición $V(P) = \{V_1, V_2, \dots, V_n\}$, generada por los puntos (p_1, p_2, \dots, p_n) que corresponden a las posiciones de los dispositivos que satisface:

$$V_i = \{q \in Q \mid \|q - p_i\| \leq \|q - p_j\|, \forall j \neq i\} \quad A.1$$

Dada una región de Voronoi se definen su masa y su centroide:

$$M_V = \int_V \rho(q) dq \quad A.2$$

$$C_v = \frac{1}{M_V} \int_V q \rho(q) dq \quad A.3$$

donde $\rho(q)$ es la función de densidad de masa de la abstracción, que determina que zonas de la región resultan más interesantes.

Cuando dos regiones de Voronoi V_i y V_j comparten una arista, se dice que los robots p_i y p_j son vecinos.

Estas particiones son las que minimizan la función de localización óptima H para la mejor percepción de los robots, evitando ruido causado por grandes distancias [5]:

$$H(P, W) = \sum_{i=0}^n \int_{W_i} f(\|q - p_i\|) \rho(q) dq \quad A.4$$

La función f corresponde con la función de percepción de los dispositivos, dependiente de la distancia entre los puntos de su partición y su posición, y ρ la que da el peso de cada punto de la abstracción como se ha visto antes.

Los centroides de cada una de las regiones de Voronoi minimizan a su vez la función H , dentro de del área de la partición, asegurando una mejor percepción de la abstracción y el óptimo recubrimiento posible del área de la abstracción [5].

$$C_{v_i} = \arg \min_{p_i} H(P, V_i) \quad A.5$$

Para calcular la región de Voronoi se procede al cálculo de las rectas que se corresponden con la mediatriz de la pareja de robot correspondiente, es decir, se ha de hallar en primer lugar la línea de puntos que están a igual distancia del robot que se ha seleccionado y sus vecinos, emparejándolos dos a dos. Se deduce por tanto que para que un robot calcule su partición ha de conocer únicamente la posición de sus vecinos y las variables de la abstracción.

Posteriormente se procede a calcular el punto de corte entre los límites del entorno, en nuestro caso una circunferencia, y las rectas halladas anteriormente.

Con esos puntos límite de la partición de cada dispositivo se obtiene el peso y el centroide de la misma, lugar hacia el cual se desplaza dicho robot con el fin de evitar golpes entre ellos. Se ha considerado una abstracción con peso constante en toda su extensión, por lo que el peso total de la partición se corresponde con el área de la zona.

Una vez hallado el diagrama de Voronoi, cada robot calcula la acción proporcional al error entre su situación actual y el centroide de su región de Voronoi, de modo que el movimiento hacia el centroide garantiza que los robots mejoran la percepción del entorno,

$$u_i = k_r (C_{v_i} - p_i) \quad A.6$$

con $k_r > 0$ la ganancia del control.

El control propuesto considera un entorno estático Q [5]. Nos encontramos en un entorno dinámico, el cual va evolucionando hasta conseguir una nueva configuración, de esta manera se puede entender que el entorno representa la abstracción que engloba al conjunto de robots. Las entradas a aplicar son proporcionales al error entre los parámetros de la formación objetivo y los actuales:

$$u_{centro} = k_{centro} (centro^* - centro) \quad A.7$$

$$u_{radio} = k_{radio} (radio^* - radio) \quad A.8$$

La constante de la entrada para los dispositivos, k_r , debe ser mayor que la de los parámetros de la abstracción, k_{centro} y k_{radio} , ya que si se produjesen cambio muy brusco en alguno de ellos los robots se podrían ver fuera su partición de Voronoi.

Los parámetros de la abstracción se actualizan mediante las siguientes ecuaciones:

$$centro(t + 1) = centro(t) + u_{centro} \Delta t \quad A.9$$

$$radio(t + 1) = radio(t) + u_{radio}\Delta t \quad A.10$$

Con cada desplazamiento de los robots se procede al cálculo del nuevo diagrama de Voronoi con referencia a situación real de la abstracción, la cual se va desplazando a la formación deseada.

El control se da por finalizado cuando todos los robots han llegado a su centroide, y se cumplan todos los parámetros de destino, es decir, el centro y el radio de la partición son correctos.

Anexo B. Conversión de velocidad

Anexo B.1 Ensayos de velocidad de simulación a comandos del robot

Para la calibración de velocidades, se desarrollaron varios ensayos experimentales en los que se han obtenido diversos resultados. La primera aproximación empírica se realizó para comandos de velocidades positivas y lineales.

Se han enviado comandos de velocidad comprendidos entre 50 y 220 durante un tiempo de 3 segundos en el caso de las velocidades bajas para observar mejor el desplazamiento realizado y de 1 o 0,5 segundos en el caso de las velocidades más altas en las que el desplazamiento realizado es más apreciable a simple vista. Se ha medido dicho desplazamiento en metros y se ha calculado la velocidad. Los resultados de este experimento se encuentran en la Tabla B.1.1 y Figura B.1.1:

Comando de velocidades positivos	Tiempo [s]	Espacio recorrido [m]	v[m/s]
50	3	0,04	0,0133
60	3	0,21	0,0700
65	3	0,23	0,0767
70	3	0,24	0,0800
80	3	0,32	0,1067
85	3	0,37	0,1233
90	3	0,38	0,1267
100	3	0,53	0,1767
105	2	0,42	0,2100
110	1	0,21	0,2100
115	1	0,23	0,2300
120	1	0,23	0,2300
130	1	0,23	0,2300
140	1	0,28	0,2800
145	1	0,28	0,2800
150	1	0,3	0,3000
160	1	0,31	0,3100
170	1	0,34	0,3400
180	0,5	0,2	0,4000
190	0,5	0,21	0,4200
200	0,5	0,22	0,4400
210	0,5	0,23	0,4600
220	0,5	0,23	0,4600

Tabla B.1.1. Comandos de velocidad positivos y velocidad lineal.

Y a partir de los valores recogidos en la Tabla B.1.1 se ha obtenido la siguiente gráfica con la siguiente expresión:

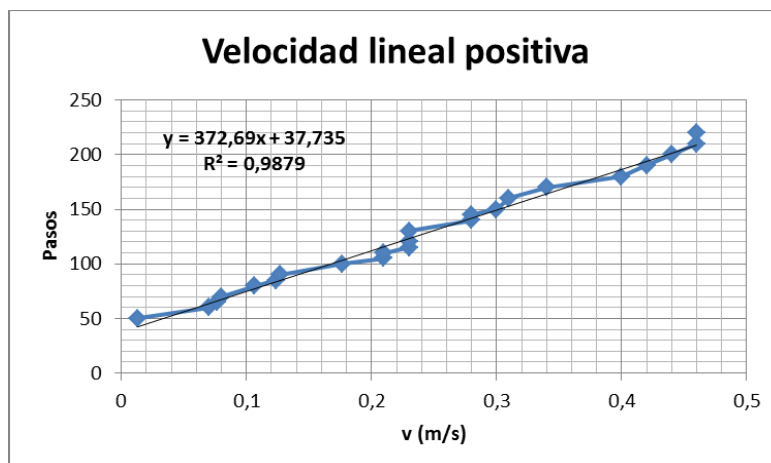


Figura B.1.1. Relación m/s y comandos de velocidad positivos

$$\text{ComandoVelPositivo} = 377,69 v + 37,735$$

B.1.1

A la vista de la gráfica, vemos que la relación comandos de velocidad-m/s es casi perfectamente lineal. Sin embargo, el primer inconveniente que se observa a la vista de la Tabla B.1.1.1 y de la Figura B.1.1 es que el rango útil de velocidad de comando comienza con el comando de velocidad igual a 50. Se ha considerado que en el rango de velocidad comprendido entre [0, 50] es igual a velocidad de 0.

El siguiente experimento se ha realizado para comandos de velocidades lineales y negativos. Como en el caso anterior, se ha vuelto a mandar comandos de velocidad al robot comprendidos entre -50 y -220 como se puede ver en la Tabla B.1.4 y se ha medido el desplazamiento realizado. En este caso, destacar que los tiempos de desplazamiento de 3 segundos se han reducido con respecto al experimento anterior dado que la velocidad lineal negativa obtenida para el comando de velocidad -50 es casi la misma que en el caso de 80 en el primer caso. A raíz de estos datos podemos asegurar que el movimiento hacia delante y hacia atrás es bastante diferente. Se puede observar dichos resultados en la Tabla B.1.2 y Figura B.1.2:

Comando de velocidades negativas	Tiempo[s]	Espacio recorrido[m]	v[m/s]
-50	3	-0,28	-0,0933
-60	3	-0,4	-0,1333
-65	2	-0,29	-0,1450
-70	1	-0,21	-0,2100
-80	0,5	-0,11	-0,2200
-85	0,5	-0,11	-0,2200
-90	0,5	-0,13	-0,2600
-100	0,5	-0,14	-0,2800
-105	0,5	-0,17	-0,3400
-110	0,5	-0,19	-0,3800
-115	0,5	-0,21	-0,4200
-120	0,5	-0,21	-0,4200
-130	0,5	-0,22	-0,4400
-140	0,5	-0,24	-0,4800

-145	0,5	-0,25	-0,5000
-150	0,5	-0,26	-0,5200
-160	0,5	-0,28	-0,5600
-170	0,5	-0,29	-0,5800
-180	0,5	-0,31	-0,6200
-190	0,5	-0,33	-0,6600
-200	0,5	-0,34	-0,6800
-210	0,5	-0,36	-0,7200
-220	0,5	-0,37	-0,7400

Tabla B.1.2. Comandos de velocidad negativos y velocidad lineal

Y a partir de los valores recogidos en la Tabla B.1.2 se ha conseguido la siguiente gráfica con la siguiente expresión:

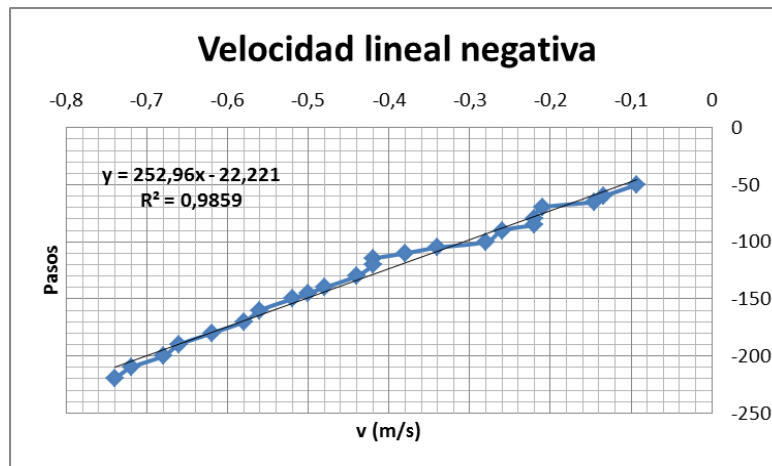


Figura B.1.2. Relación m/s y comandos de velocidad negativos

$$\text{ComandoVelNeg} = 252,98v - 22,21$$

B.1.2

En cuanto a semejanzas entre la Figura B.1.1 y la Figura B.1.2, volvemos a encontrar que la relación comandos de velocidad-m/s es casi lineal y que existe un rango de velocidades obsoleto comprendido entre $[-50, 0]$ en el cual hemos vuelto a considerar velocidad lineal igual a 0.

El siguiente experimento trata de comprender el movimiento angular de los robots. Se ha comenzado por los comandos de velocidad angular positivos. Se han enviado comandos de velocidad comprendidos entre 130 y 255 durante un tiempo de 2 segundos. Se han contado las vueltas dadas en ese intervalo de tiempo y se ha calculado la velocidad angular. Los datos obtenidos se muestran en la Tabla B.1.3 y Figura B.1.3:

Comandos de velocidad angular positiva	Tiempo [s]	Vueltas	w [rad/s]
130	2	0,27	0,8482
140	2	0,3	0,9425
150	2	0,6	1,8850
160	2	0,6	1,8850
170	2	0,75	2,3562
180	2	0,75	2,3562
190	2	0,8	2,5133
210	2	1,1	3,4558
220	2	1,15	3,6128
240	2	1,2	4,0000
255	2	1,23	4,7000

Tabla B.1.3. Comandos de velocidad positivos y velocidad angular

Y a partir de los valores recogidos en la Tabla B.1.3 se ha obtenido la siguiente gráfica con la siguiente expresión:

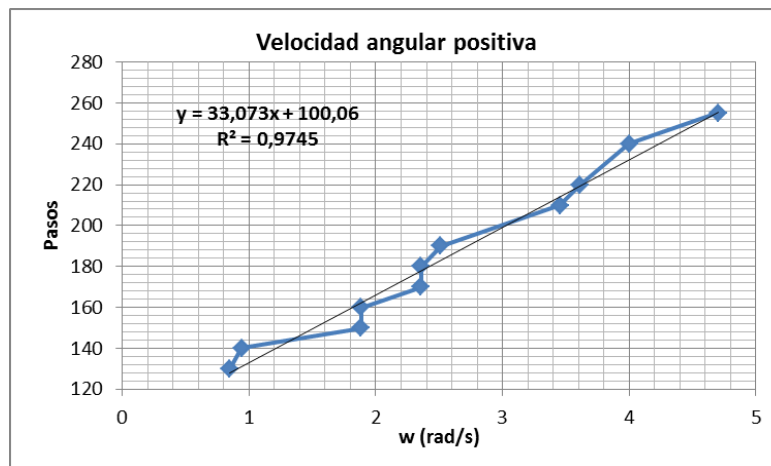


Figura B.1.3. Relación rad/s y comandos de velocidad positivos

$$\text{ComandoWPos} = 33,073w + 100,06$$

B.1.3

Se aprecia como la relación comandos de velocidad-rad/s es prácticamente lineal en la Figura B.1.3. Sin embargo, vuelve a aparecer el inconveniente de los casos anteriores: se observa que el rango útil de velocidad de comando comienza con el comando de velocidad igual a 130. Se ha vuelto a considerar que en el rango de velocidad comprendido entre [0, 130] es igual a velocidad de 0. También cabe destacar que el movimiento con comando de velocidad angular positivo es en el sentido de las agujas del reloj.

A continuación, se procede a analizar el movimiento con comandos de velocidad angular negativas. Como ocurría en el caso de velocidad angular positiva, podemos ver como se han enviado comandos de velocidad comprendidos entre -130 y -255 durante un intervalo de tiempo de 2 segundos. Se han contado las vueltas dadas en ese intervalo de tiempo y se ha calculado la velocidad angular. Los resultados se plasman en la Tabla B.1.4 y Figura B.1.4:

Comandos de velocidad angular negativa	Tiempo[s]	Vueltas	w [rad/s]
-130	2	0,25	-0,7854
-140	2	0,4	-1,2566
-150	2	0,5	-1,5708
-160	2	0,6	-1,8850
-170	2	0,65	-2,0420
-180	2	0,7	-2,1991
-190	2	0,8	-2,5133
-210	2	0,9	-2,8274
-220	2	1	-3,1416
-230	2	1,1	-3,4558
-240	2	1,2	-3,7699
-255	2	1,3	-4,0841

Tabla B.1.4. Comandos de velocidad negativos y velocidad angular

Y a partir de los valores recogidos en la Tabla B.1.4 se obtuvo la siguiente gráfica con la siguiente expresión:

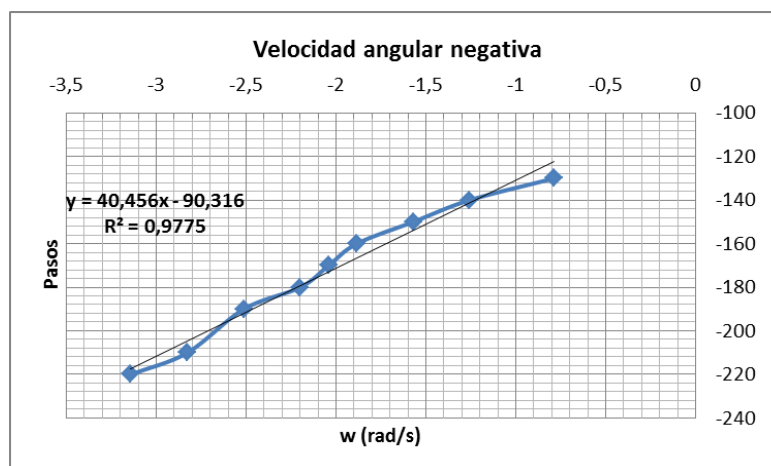


Figura B.1.4. Relación rad/s y comandos de velocidad negativos

$$\text{ComandoWNeg} = 40,0456w - 90,316$$

B.1.4

La relación comandos de velocidad-rad/s es casi lineal como se puede ver en la Figura B.1.4. De nuevo, aparece el inconveniente de los casos anteriores donde el rango útil de velocidad de comando comienza con el comando de velocidad igual a -130, Tabla B.1.4. Se ha vuelto a considerar que en el rango de velocidad comprendido entre [-130, 0] es igual a velocidad 0. Como cabe esperar, si anteriormente el sentido de giro es en sentido de las agujas del reloj, en este caso es al revés, es decir, al contrario de las agujas del reloj

Para el caso de velocidades angulares, vemos que las rectas que relacionan w [rad/s] y comandos de velocidad angular son algo más similares que en el caso de velocidades lineales.

Anexo B.2 Análisis de velocidades de comando a motor derecho e izquierdo

En cuanto a la velocidad transmitida a las ruedas, se observan problemas de no linealidades de movimiento derivado de los rangos en los que la velocidad es nula, muy baja o muy alta. Dada esta situación, ha sido necesario realizar un ajuste proporcional de velocidad lineal y velocidad angular para que los robots describieran correctamente sus movimientos. Por ello, se han implementado unas constantes proporcionales a v y w dependientes de la velocidad en cada instante:

$$(v, w) = f(k_v, k_w) \quad \text{B.2.1}$$

De este modo, dados unos valores límites estudiados experimentalmente de velocidad lineal y angular (e_v y e_w) utilizados para discriminar entre movimiento rotacional, movimiento traslacional, movimiento a baja velocidad y movimiento a alta velocidad. Estos valores son independientes para cada uno de los robots y se han considerado diferentes situaciones:

- Si $v < |e_v|$ y $w > |e_w|$: se considera que el movimiento descrito es rotacional, por lo que k_v disminuye y k_w aumenta, de esta manera el movimiento es un giro.

$$v_R = k_v \text{ComandoV} - k_w D \text{ComandoW} \approx -k_w \text{ComandoW} \quad \text{B.2.2}$$

$$v_L = k_v \text{ComandoV} + k_w D \text{ComandoW} \approx +k_w \text{ComandoW} \quad \text{B.2.3}$$

- Si $v > |e_v|$ y $w < |e_w|$: se considera que es un movimiento es traslacional, por lo que k_v aumenta y k_w disminuye, de esta manera el movimiento es un desplazamiento lineal.

$$v_R = k_v \text{ComandoV} - k_w D \text{ComandoW} \approx k_v \text{ComandoV} \quad \text{B.2.4}$$

$$v_L = k_v \text{ComandoV} + k_w D \text{ComandoW} \approx k_v \text{ComandoV} \quad \text{B.2.5}$$

- Si $v < |e_v|$ y $w < |e_w|$: se considera que el movimiento es mínimo, con lo que los valores de k_v y k_w son igual a 0.

$$v_R = k_v \text{ComandoV} - k_w D \text{ComandoW} = 0 \quad \text{B.2.6}$$

$$v_L = k_v \text{ComandoV} + k_w D \text{ComandoW} = 0 \quad \text{B.2.7}$$

- Si $v > |e_v|$ y $w > |e_w|$: se considera que el movimiento es un desplazamiento con giro, con lo que los valores de k_v y k_w no varían. La velocidad máxima a proporcionar son 255 o -255, por lo que v_R y v_L no superará dichos valores:

$$v_R = k_v \text{ComandoV} - k_w D \text{ComandoW} \quad \text{B.2.8}$$

$$v_L = k_v \text{ComandoV} + k_w D \text{ComandoW} \quad \text{B.2.9}$$

Anexo C. Resultados Experimentales

Experimento 2 con identificación visual

En cuanto al experimento 2, también se han propuesto dos objetivos: uno de ellos con centro igual a $(2\text{m}, 1,5\text{m})$ y radio 1m , y el segundo $(0,75\text{m}, 1\text{m})$ y radio $0,75\text{m}$. La distribución de colores en lo que se refiere a objetivos, abstracciones y posiciones es la misma que en el caso anterior.

Atendiendo a la figura C.1, se puede observar la configuración inicial, de nuevo aleatoria, el desplazamiento de los robots y el alcance de todos los robots a su primer objetivo.

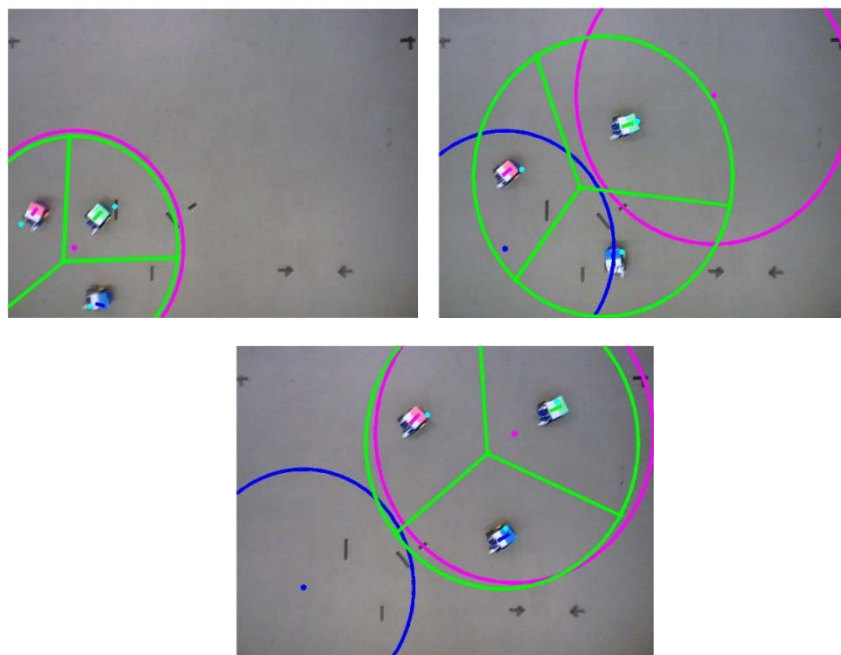


Figura C.1. Robots en bucle cerrado alcanzando primer objetivo.

En la Figura C.2 se observa como se reduce el tamaño de la abstracción para poder alcanzar con éxito el segundo objetivo. Volvemos a observar como el robot de etiqueta verde paraliza el movimiento de la abstracción hasta que se reorienta y se dirige a su centroide. Todos los robots siguen a su centroide y logran su objetivo final con éxito.

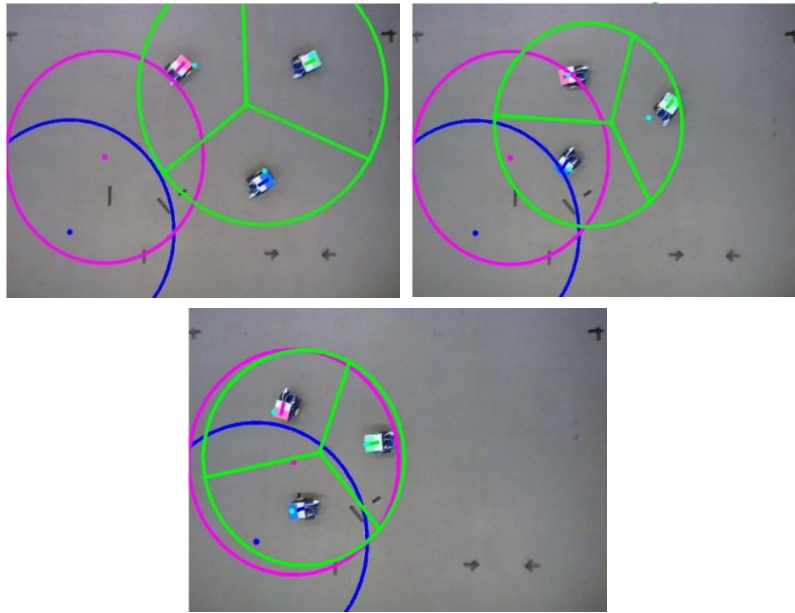


Figura C.2. Robots en bucle cerrado alcanzando segundo objetivo.

De nuevo, aparecen errores de localización como se puede apreciar en la Figura C.3. Se observa como el robot de etiqueta azul no es identificado, probablemente se debe a que el área de color reconocida era algo más pequeña y como consecuencia ha sido tratada como un ruido y, por tanto, eliminada. La consecuencia se plasma visualmente en las particiones calculadas por el control. Si la comparamos la Figura C.3 con la C.2 se ve como las particiones calculadas en un caso y en el otro difieren significativamente. Sin embargo, el problema ha sido resuelto rápidamente y la formación ha logrado su objetivo final.

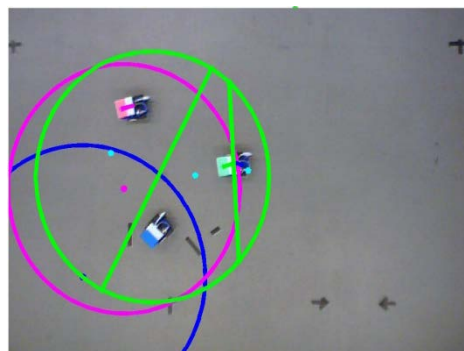


Figura C.3. Problema de reconocimiento.

En cuanto a las velocidades, a la vista de las gráficas de la Figura C.5 vuelven a existir grandes variaciones. Los valores de acción no superan al establecido como máximo. En la gráfica de acción de velocidad, se observa como las variaciones son bastantes. En cuanto a la gráfica de la parte inferior que muestra el error de las posiciones de los robots con respecto a su centroide objetivo, se puede ver como sí se supera en más ocasiones el error de 0,10m, llegando incluso en alguna ocasión a los 0,25m, lo que equivale en un desplazamiento del 12% en un desplazamiento de dos metros. Pese a esto, los errores terminan reduciéndose cuando los robots llevan llegando a su objetivo final y, por tanto, se puede ver como en la mayoría de las

ocasiones no se supera los 10 cm de error, es decir, el error en casi todo el recorrido es de entorno al 7%.

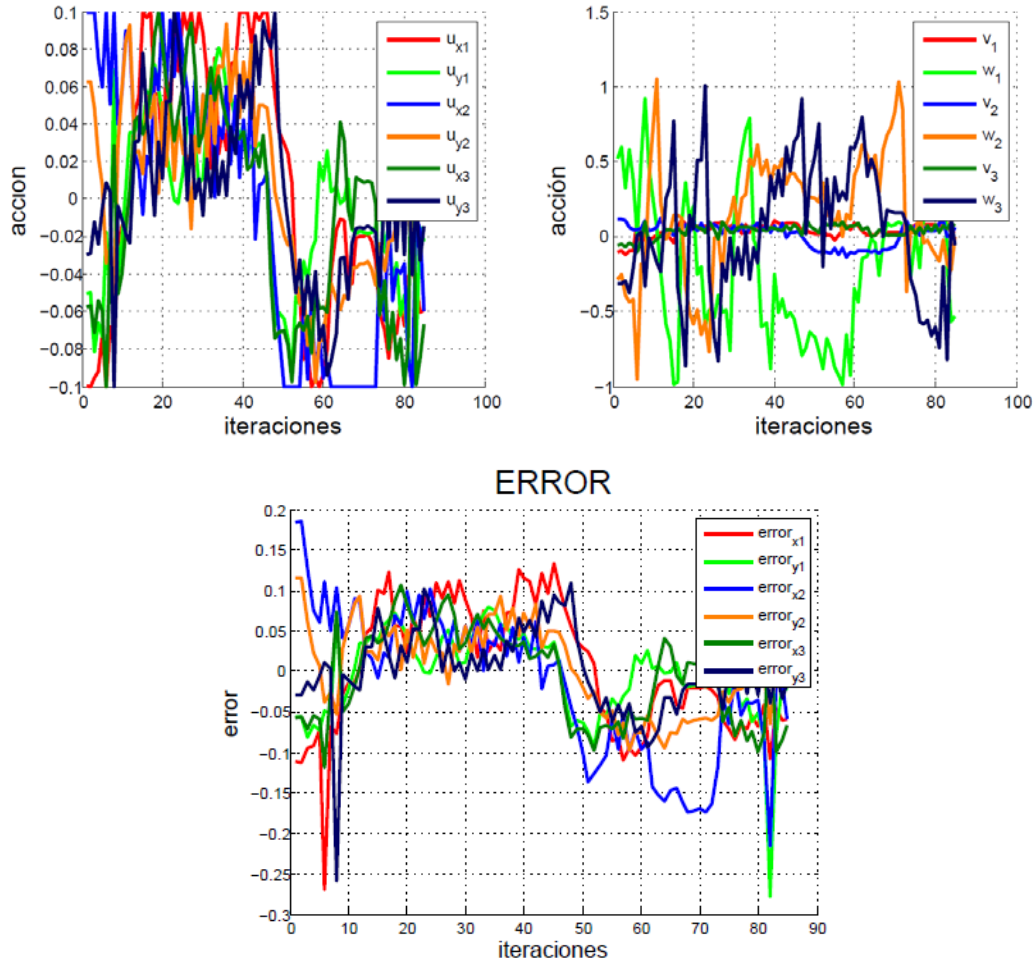


Figura C.5. Acción para $N = 3$ centro = $(2m, 1,5m)$ y radio = $1m$ y centro $(0,75m, 1m)$ y radio = $0,75m$. Error de posición en x e y para cada robot con respecto al centroide.